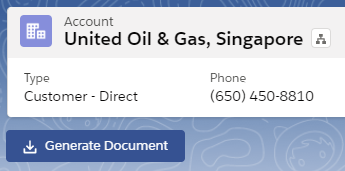
# Salesforce Setup – Lightning version

This is guide for setting up a connector to our Document Automation API. If you are interested in a trial reach out to the API team at [api@skabelondesign.com](mailto:api@skabelondesign.com)

By completing the guide, you will be able to make API requests to SkabelonDesign’s web service.



Every code example can be found in our public GitHub repository:

<https://github.com/sebastianehlers/skabelondesign-api-connectors/tree/master/salesforce/lightning/code>

Contents

[Salesforce Setup – Lightning version 1](#_Toc23843412)

[1 Pre-requisites 2](#_Toc23843413)

[2 API Remote Site Settings 2](#_Toc23843414)

[3 Creating domain 3](#_Toc23843415)

[4 Connector setup 4](#_Toc23843416)

[4.1 Button setup 6](#_Toc23843417)

[4.2 Button controller setup 7](#_Toc23843418)

[4.3 Document Generation Controller – Apex class 8](#_Toc23843419)

[5 Editing page and placing the button 9](#_Toc23843420)

[6 Configuration 12](#_Toc23843421)

[6.1 Authentication token 12](#_Toc23843422)

[6.2 Service Host 12](#_Toc23843423)

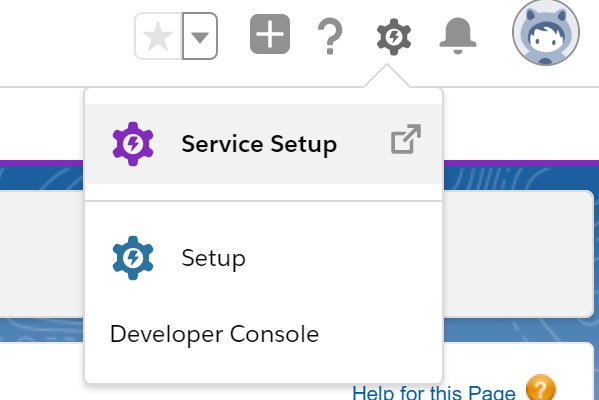
[6.3 Template Id 12](#_Toc23843424)

## Pre-requisites

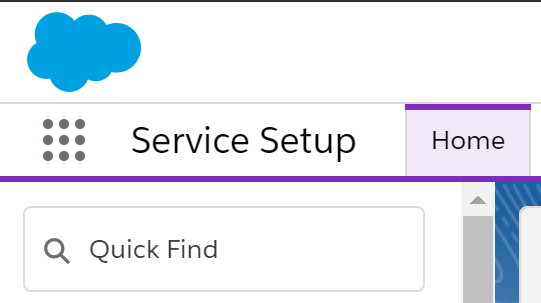
1. System Administrator access to a salesforce tenant.
2. Custom domain for your Salesforce URL. Setup guide -> Creating domain**.**
3. Test tenant to make successful requests to the API – can be obtained by request to [api@skabelondesign.com](mailto:api@skabelondesign.com)

## API Remote Site Settings

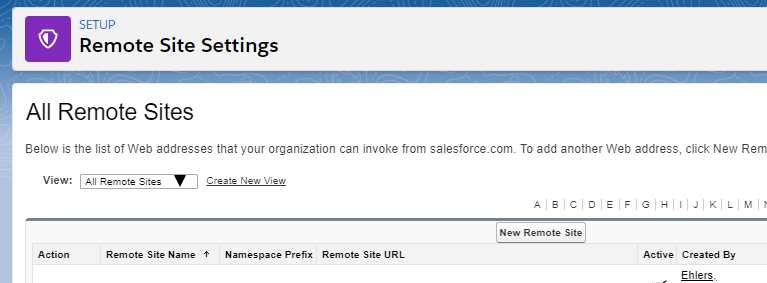
Before we start setting up the connector, create new remote site that can be invoked from Salesforce. This can be done by first going to “Service Setup”.



And then by searching for “Remote Site Settings” in the “Quick Find” to the left page side:



Create a new remote site by clicking on “New Remote site” as shown on the image below.



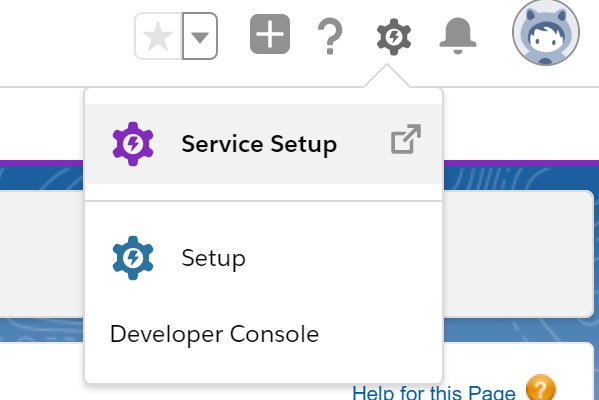
Fill out the *Remote Site Name* and *Remote Site URL* fields and press on save.

**Remote Site Name** – up to the configurator.

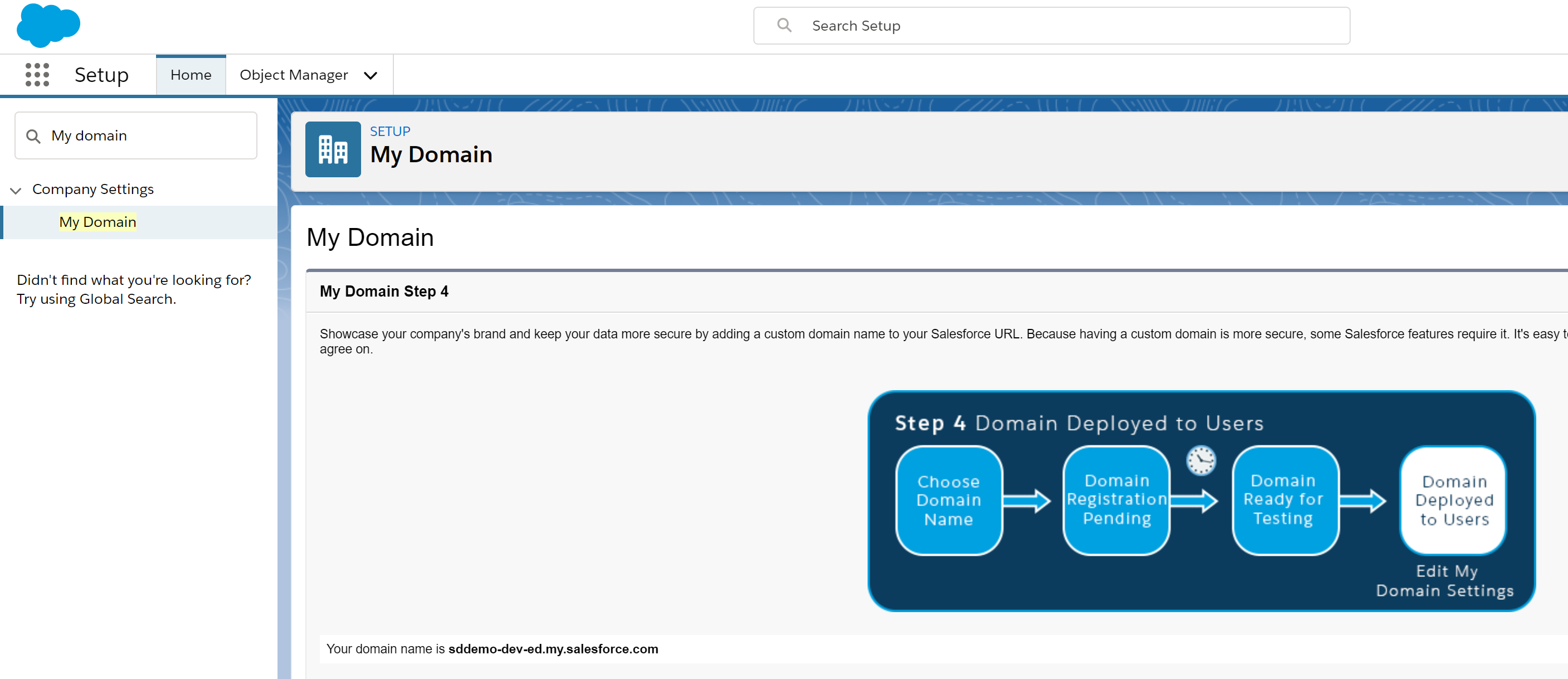
**Remote Site URL -** https://templafy.officeextensions.net/

## Creating domain

Go to “Service Setup” that can be found here:



In “Quick find”, please type “My domain”.



Follow steps to setup a custom domain. When finished, you should be able to see the “My Domain” view as shown above + “Your domain name is ...” in the view on the right side. Remember to activate the domain for all users.

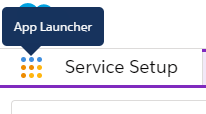
## Connector setup

The setup of the controller will consist of four steps:

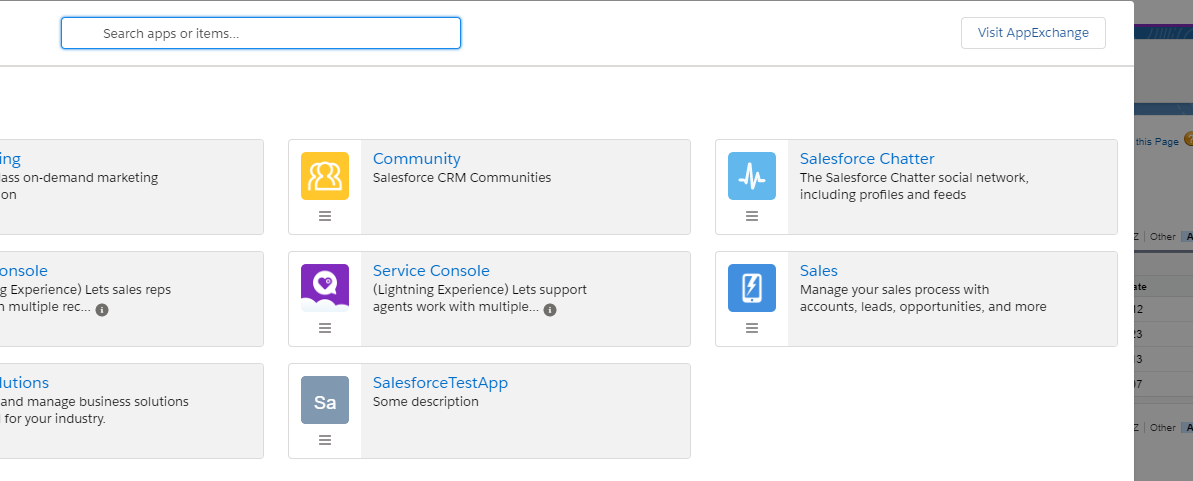
1. Button setup (View)
2. Button controller (Js controller)
3. Document Generation Controller (Apex class)
4. Editing page view and placing button

Before we can start configuring the above steps, we need to be in the Developer Console. We can achieve that by following the guide underneath.

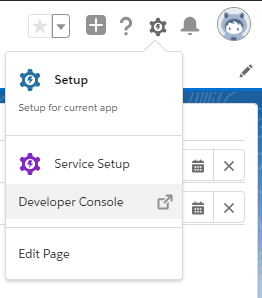
Please go to the “Sales” application in your Salesforce tenant. You can achieve that by clicking on the following dotted brick:



And then select “Sales”.

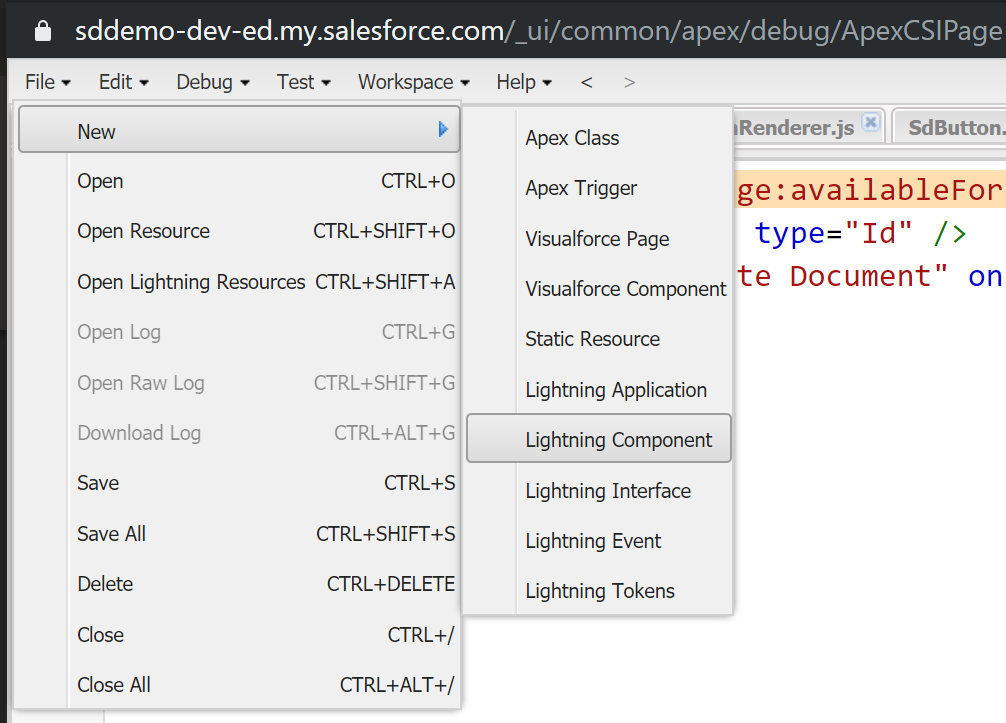


Select then “Developer Console” as shown below (top right corner):



### Button setup

Create a new component by selecting File -> New -> Lightning Component as show below and define its name to be **SdButton**.



The button design is up to the configurator. Official documentation regarding styling can be found here: <https://developer.salesforce.com/docs/component-library/bundle/lightning:button/example>

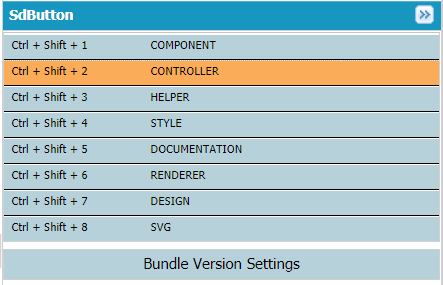
Copy the sample button code + necessary setup to call the function from:

<https://github.com/sebastianehlers/skabelondesign-api-connectors/blob/master/salesforce/lightning/code/SdButton.cmp>

and save the file.

### Button controller setup

After making the button, we need to configure some dynamics. In order to do that, click on the “Controller” as indicated on the screenshot below.



Paste the following example code for the sample controller from:

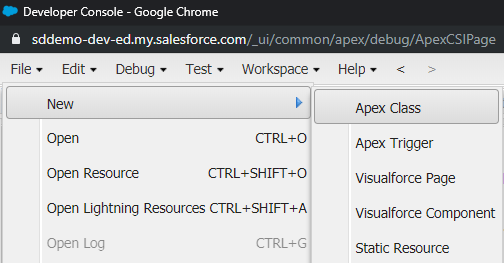
<https://github.com/sebastianehlers/skabelondesign-api-connectors/blob/master/salesforce/lightning/code/SdButtonController.js>

and save the file.

### Document Generation Controller – Apex class

The last step is to create an Apex class (serve-side class) that will execute SOQL queries as well as make API requests to SD’s endpoint.

Create a new Apex class by selecting File -> New -> Apex class as shown below.



Name the class as: **DocumentGenerationController**

The binding to the controller can be seen on the button and therefore the name needs to match 1:1. Binding can be seen on the first line of the button config (controller="DocumentGenerationController").

Copy the code from:

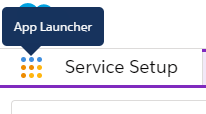
<https://github.com/sebastianehlers/skabelondesign-api-connectors/blob/master/salesforce/lightning/code/DocumentGenerationController.apxc>

and save the file.

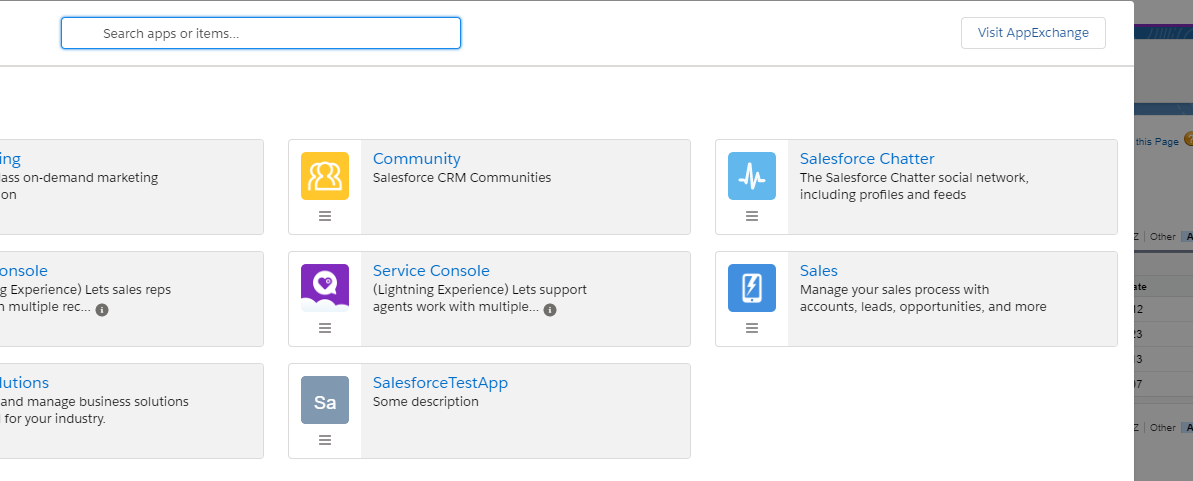
## Editing page and placing the button

Last step is to place the button on your Salesforce page.

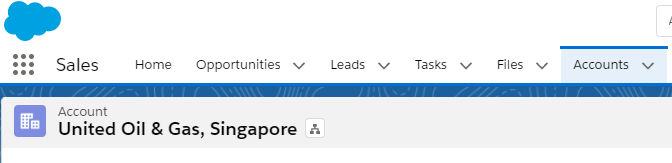
You can achieve that by clicking on the following dotted brick:



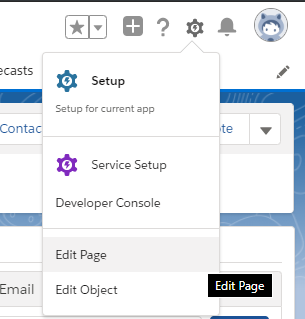
And then select “Sales”.



Go to the *Accounts* tab and select one of the accounts. You will now be able to see an account with its details as shown below.

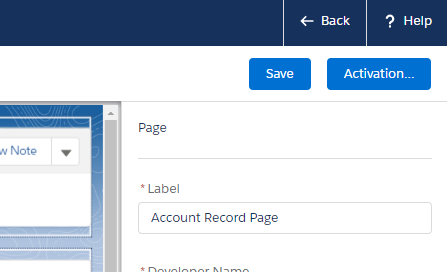


Go to the top right corner and select “Edit Page” from the drop down as indicated on the picture below.



You will now be redirected to the page editing.

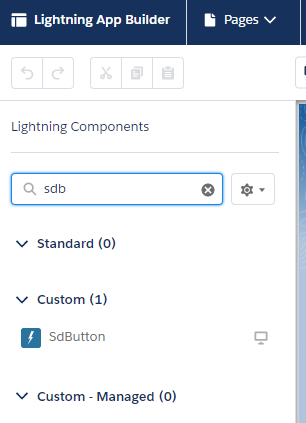
Let’s activate the edited page to be the App Default. You can do that by clicking on Activation button to the right.



And then by clicking on “Set as Org Default”. Close the view.

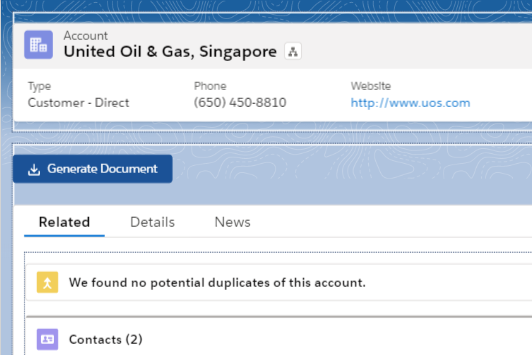
You will now be able to see the custom components in your view to the left side.

You can search for your button by name and see the custom component as shown on the image below:



Drag the button to the page (up to the configurator) and click on “Save” to the right.

You should now be able to see the button on the edited page as well as in the final page view.



Go “Back” to the view and select Accounts -> Some Account and start generating documents by pressing on “Generate Document” button.

## Configuration

The connector example requires three inputs in order to function:

* Authentication token
* Service host tenant
* Id of the template to use when generating documents

Contact the SkabelonDesign API team to receive this information and instruction on configuration: [api@skabelondesign.com](mailto:api@skabelondesign.com)

### Authentication token

In DocumentGenerationController.apxc file modify line 3 and replace “token here” with your provided authentication token.

### Service Host

In DocumentGenerationController.apxc file modify line 2 and replace “host here” with your provided host.

### Template Id

In SdButtonController.js file, modify line 4 and replace “template id here” with your provided template id.